

RADAR: A Fast, Scalable, and Distributable Archive Inventory Service

Anzhen Zhang, Thomas H. Jarrett, Anastasia Alexov, G. Bruce Berriman, John C. Good, Mihseh Kong, Naveed D. Tahir-Kheli, and Serge Monkewitz

Infrared Processing and Analysis Center, California Institute of Technology, Mail Code 100-22, CA 91125

Abstract. The NASA/IPAC Infrared Science Archive (IRSA) has recently deployed the Recursive Archive Digest and Reference (RADAR) service, which returns an inventory of IRSA's holdings in response to a spatial query, and offers one-click download of data and links to IRSA's data access services. RADAR also supports inventories and data access from remote archives; the current implementation supports access to the Multi-mission Archive at STScI (MAST) Spectral and Image Scrapbook and NEDBasic Data. When complete, RADAR will maintain the results of multiple queries in "data collections" and will provide tools that will allow users to augment collections, remove data from them, modify search criteria, resubmit jobs, and check job status. RADAR is supported by an evolution of IRSA's component based architecture. It utilizes a fast estimation service and runs under the Request Management Environment (ROME) funded by NVO.

1. Introduction

RADAR (Recursive Archive Digest and References; <http://irsa.ipac.caltech.edu/applications/RadarSvc>) was developed by the NASA/IPAC Infrared Science Archive (IRSA) to return inventories of the archive's growing holdings in response to a spatial query. It supports fast inventories of the archive holdings, generation of multiple inventories in the same browser session, one-click packaging and download of data, and links to custom services for individual datasets to support further exploration of the data.

This paper describes the design challenges faced by RADAR, and how they were met to deploy an easily maintainable service that retains its performance as the archive holdings grow.

2. Design and Architecture

RADAR serves as a proxy application, through which the user can make complex data retrieval requests to IRSA's public datasets. IRSA is a multi-mission archive. It hosts huge source catalogs which are stored in the database, and smaller source catalogs that reside on spinning disk. It curates large image datasets and smaller spectral datasets, which are stored on spinning disks while their metadata are stored inside the database; some of these data are in FITS format, while others are JPEG or PostScript files. The archive thus curates a

broad range of data formats, data storage, and data access mechanisms. Providing generic software which manages these different data format and access mechanisms in a manner that appeared uniform to the user was a major challenge in designing RADAR.

In addition to the diversity of the holdings, IRSA frequently updates its data holdings and releases new datasets according to data providers' delivery schedules. Some of them may be updated several times a year. IRSA deploys custom interfaces for all new datasets, and updates existing interfaces when data are updated. RADAR thus needs to respond dynamically to updates to the data holdings, while at the same time incorporating inventories of new datasets.

Based on those considerations, RADAR is driven by a master table of datasets. Any updates to an existing dataset only require a modification to its data pointers in the table. Every new dataset release can be incorporated by adding one entry in the master table. This design provides flexibility in supporting updates to the archive holdings and minimizes the additional development and test burden needed to support them.

RADAR is accessible as a web form. It receives requests through a web server (Apache) which are handed off to C-based CGI programs called "applications". Separation of the processing into applications is overkill for RADAR, where coordinated processing across a set of applications / datasets was desired, either to collect data from various sources in response to a higher level request (e.g., for an area on the sky) or as part of a distributed processing paradigm. RADAR is meant to run data access applications simultaneously. Therefore, RADAR has adapted middleware called the Request Management Environment (Kong, Good & Berriman 2004) to take advantage of multi-threaded processing of applications and provide scalable performance as the archive holdings grow.

Most IRSA applications are CGI programs that can accomplish a request in real time. They usually respond with HTML (generated by the CGI program) encapsulating or referencing the data which was generated in the workspace. RADAR requires that every application provide a mode which performs this same processing without standard output; instead, it generates a resultant output file in HTML which will be used by the RADAR Collection Viewer (see Section 6). In this mode, each application provides status information in XML format, which facilitates communication between RADAR, ROME, and itself.

Some applications may take some time to complete (such as long database queries, 2MASS images and, in future, custom image mosaics). In those cases, the applications provide their own status pages. Those pages provide detailed information about every process step and its status in real time. A polling mechanism is in place to check status via a browser. Similar status messages are also passed to the RADAR Collection Viewer through ROME during data processing.

Figure 1 shows a functional flow diagram for RADAR. It shows the relationship between RADAR components and IRSA data access applications. The RADAR components are described in the following sections.

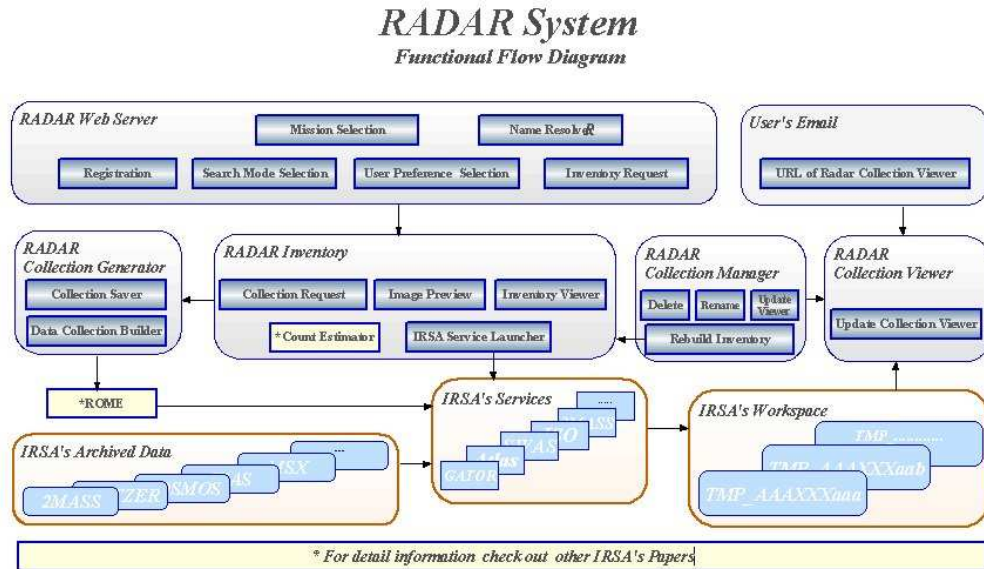


Figure 1. Functional Flow Chart for RADAR

3. Data Collection Generator

A data “collection” in RADAR is defined as any set of data, metadata, references to data, ancillary information, and the organizational structure associated with storing the data for a specific spatial area or target. The collection names must be unique, but otherwise have no naming restrictions. An example of a collection can be all the images, archives, spectra, and catalog subsets associated with a region on the sky plus the metadata describing these data (e.g., image coverage information), arranged into a coherent directory tree.

A collection actually consists of a variety of files. They can be data files or control files for processes run by RADAR, or information and status files which detect the health of software in IRSA. The Collection Generator is responsible for generating a set of control and information files. They record various properties and parameters of a collection. As one example, RADAR derives information from those files and automatically fills in the object name, query criteria, selected datasets, email, and work space information from the last job for a user. Another example is JOBLIST, one of the job control files in XML, which defines a list of datasets for a collection. At RADAR’s request, ROME utilizes this file and starts each IRSA CGI program on the list and sends the process status back to RADAR.

The Collection Generator is what makes simultaneous access to multiple datasets possible.

4. Inventory

The Inventory service performs several functions. It checks for data availability at IRSA and quickly returns an inventory of the holdings. The latter utilizes a

fast estimation algorithm by taking advantage of pre-calculated statistics on the distribution of sources and image and spectral data across the sky. For catalogs and spectra, the source counts are exact sources, but for some images, such as 2MASS Atlas, the counts are approximate because we assume that the images are point sources found in real database queries. For some datasets where IRSA does not have access to pre-computed statistics, the inventory only indicates that sources are found in the search area (e.g., ISO).

The inventory provides two ways to access data from IRSA. The first is one-click access to datasets selected from the results page, which runs the Collection Generator to perform multiple data access calls simultaneously. When processing is complete, the data are bundled and made available for pick up. The second way is run individual data access services through a browser, specific to a collection or individual application. For every service, the inventory automatically fills index pages with the search criteria input to RADAR. The first method is best for getting a large volume of data simultaneously; the second method is best for data exploration.

Although the inventory return page only lists holdings where data are found, a complete list of all datasets, with or without data meeting the search criteria, is also available.

5. Collection Manager

The Collection Manager is designed to generate and manipulate data collections created earlier. When the Collection Manager is displayed, the user's profile is on the top and all collections belonging to that user are listed. Each collection can be examined simply by clicking on the collection name, and the collection can be updated by entering new search criteria.

6. Data Collection Viewer

The Collection Viewer is responsible for generating a presentation of a data collection in a HTML page suitable for access by a scientist. It summarizes and organizes all the sources which have been found by RADAR together with metadata describing the datasets from which the collection is derived.

The Viewer supports the same data access methods as the RADAR return page: one-click data retrieval and access to IRSA web services, with search parameters filled in.

The collection name is identified at the top of the Collection Viewer, as are the query criteria. During the data retrieval process, the Collection Viewer allows users check the status of processes as they run. This update is accomplished by a set of XML files; some of them generated by RADAR, some of them generated by ROME, and others generated by each application.

7. Summary

RADAR provides a mechanism for generating inventories of input regions quickly, downloading data, and exploring the IRSA archive holdings. The application is

highly scalable. It is also distributable and can be extended to provide inventories of remote data sets. Indeed, it already supports a fast inventory of the MAST Spectral and Image Scrapbook. It also supports multiple inventories in a single browser session. When complete, it will maintain these multiple inventories in a permanent workspace assigned to individual users, and will support visualization and update of these “collections.”

References

- Kong, M., Good, J.C., & Berriman, G.B. 2004, in ASP Conf. Ser., Vol. 347, ADASS XIV, ed. P. L. Shopbell, M. C. Britton, & R. Ebert (San Francisco: ASP), 213